08/852,158
.5
ST.JC

WEST Search History

DATE: Tuesday, March 11, 2003

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L48	l32 and l44	6	L48
L47	l32 and l44	6	L47
L46	l32	2068	L46
L45	l43 and L44	1	L45
L44	memory same usage same threshold$1	346	L44
L43	l32 same (application$1 or task$1 or program$1 or process$2)	612	L43
L42	l32	2068	L42
DB=USPT; PLUR=YES; OP=OR			
L41	5572694.pn.	1	L41
L40	5630097.pn.	1	L40
L39	5727178.pn.	1	L39
L38	6078942.pn.	1	L38
L37	6078942.pn.	1	L37
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L36	windows same l32	28	L36
L35	l9 and L34	2	L35
L34	l8 and l32	329	L34
L33	critical adj memory adj threshold$1	0	L33
L32	memory adj monitor$3	2068	L32
L31	memory	1268140	L31
DB=USPT; PLUR=YES; OP=OR			
L30	5412798.pn.	1	L30
L29	5579529.pn.	1	L29
L28	5600840.pn.	1	L28
L27	5706407.pn.	1	L27
L26	5768568.pn.	1	L26
L25	5822600.pn.	1	L25
L24	5860125.pn.	1	L24
L23	5867714.pn.	1	L23
L22	5920728.pn.	1	L22
L21	5935228.pn.	1	L21
L20	5951684.pn.	1	L20

L19	6009480.pn.	1	L19
L18	6078921.pn.	1	L18
DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L17	l12 and l15	1	L17
L16	l10 and L15 and l9 and l7	0	L16
L15	l1 or l2	2982	L15
L14	l3 and l12	0	L14
L13	l7 and L12	10	L13
L12	l9 and l8	288	L12
L11	l9 and l8L10	0	L11
L10	virtual adj memor$3	6549	L10
L9	windows adj CE	877	L9
L8	(kill$1 or terminate$1 or clos$2) same (application$1 or program$1 or process$2 or task$1)	440753	L8
L7	(minimiz$3 or reduc$3) adj memory	12775	L7
L6	l4 and L5	0	L6
L5	l3	10	L5
DB=USPT; PLUR=YES; OP=OR			
L4	clos$2 same L3	0	L4
L3	application$1 same memory same allocation same threshold$1	10	L3
L2	((711/170 |711/171 |711/172 |711/173)!.CCLS.)	1300	L2
L1	((711/1 |711/2 |711/3 |711/4 |711/5 |711/6)!.CCLS.)	1860	L1

END OF SEARCH HISTORY

WEST Search History

```
DATE:  Friday, February 21, 2003
```

ST.JC

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=OR			
L9	l7 and L8	0	L9
L8	low adj memory	1408	L8
L7	l1 and l5 and L6 and l4 and l3	27	L7
L6	application adj program$1	24805	L6
L5	operating adj system	53156	L5
L4	l1 and L3	96	L4
L3	((711/170 |711/171 |711/172 |711/173)!.CCLS.)	1282	L3
L2	critical adj memory adj threshold	0	L2
L1	memory same threshold$1	27448	L1

```
END OF SEARCH HISTORY
```

08/852,158
STJC
2-21-03

End of Result Set

□ Generate Collection

L45: Entry 1 of 1   File: PGPB   Mar 6, 2003

PGPUB-DOCUMENT-NUMBER: 20030046581
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20030046581 A1

TITLE: System and method for protecting computer device against overload via network attack

PUBLICATION-DATE: March 6, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Call, R. Christian	Aberdeen	NJ	US	
Cavuto, David J.	Edison	NJ	US	
Giorgis, Tadesse W.	Neptune	NJ	US	

APPL-NO: 09/ 941981 [PALM]
DATE FILED: August 29, 2001

INT-CL: [07] G06 F 11/30

US-CL-PUBLISHED: 713/201
US-CL-CURRENT: 713/201

REPRESENTATIVE-FIGURES: 4

ABSTRACT:

The present invention protects network devices from overload and from network packet flood attacks (such as Denial of Service and Distributed Denial of Service attacks) that would otherwise consume available resources, and possibly cause system failure or compromise the system by allowing intrusion. The invention, termed an intelligent cache management system is used to free allocated resources (memory, in particular) for reuse, when under sustained attack. One exemplary embodiment of a cache management system of the present invention is used in connection with session-type packet processing devices of a computer network. The system comprises a memory management database for storing communication traffic classification and memory threshold values, and a memory monitor for tracking overall memory usage and determining when the memory threshold values stored in the memory management database are reached. A cache classifier is used to determine a class into which a given session of communications traffic falls. When the memory threshold value is reached, a pruning mechanism selects and prunes entries representing sessions on the packet processing device in accordance with the communication traffic classification and memory thresholds programmed in the memory management database.

□ [Generate Collection]

L36: Entry 11 of 28　　File: USPT　　Jun 12, 2001

US-PAT-NO: 6247042
DOCUMENT-IDENTIFIER: US 6247042 B1

TITLE: Method and system for restoring the state of physical memory as the focus changes among application programs in a computer

DATE-ISSUED: June 12, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Engstrom; G. Eric	Kirkland	WA		
Eisler; Craig G.	Kirkland	WA		

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Microsoft Corporation	Redmond	WA			02

APPL-NO: 08/ 936358 [PALM]
DATE FILED: September 24, 1997

NOT PRIOR ART

INT-CL: [07] G06 F 9/00

US-CL-ISSUED: 709/107; 711/202
US-CL-CURRENT: 709/107; 711/202

FIELD-OF-SEARCH: 709/100, 709/101, 709/102, 709/105, 709/106, 709/107, 711/130, 711/133, 711/117, 711/206, 711/165, 711/202, 711/13, 711/129, 711/209, 714/42, 714/38, 714/718

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

[Search Selected] [Search ALL]

	PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
□	4688167	August 1987	Agarwal	
□	4967353	October 1990	Brenner et al.	
□	5125086	June 1992	Perazzoli, Jr.	
□	5386536	January 1995	Courts et al.	711/136
□	5394537	February 1995	Courts et al.	711/202
□	5499354	March 1996	Aschoff et al.	711/129
□	5572694	November 1996	Uchino	709/1
□	5606685	February 1997	Frandeen	711/117
□	5611064	March 1997	Maund et al.	711/209
□	5630097	May 1997	Orbits et al.	395/492
□	5727178	March 1998	Pletcher et al.	711/202
□	6078942	June 2000	Eisler et al.	709/100

FOREIGN PATENT DOCUMENTS

FOREIGN-PAT-NO	PUBN-DATE	COUNTRY	US-CL
620 523 A3	October 1994	EP	
620 523 A2	October 1994	EP	
713 176 A3	May 1996	EP	
713 176 A2	May 1996	EP	

OTHER PUBLICATIONS

PCT/US98/16800--International Search Report, Sep. 12, 1998.
"Method of Extending OS/2's Memory management to Recognize "User Focus"", IBM Technical Disclosure Bulletin, vol. 35, No. 1A, Jun. 1992, pp. 470-472.
PCT/US98/16802--International Search Report, Sep. 12, 1998.
V. Sohal, Reliable Memory Management for Real-Time Systems, Electronic Design, vol. 44, No. 13, Jun. 1996, pp. 118, 120, 122, 124 XP000625394, see p. 120, right-hand column, line 7-page 124, left-hand column, line 39.
PCT/US98/16800--International Search Report, Jan. 22, 1999.
"Packing Variable-Sized Segments in the Swap File of a Paging-Based Virtual Memory System", IBM Technical Disclosure Bulletin, vol. 39, No. 3, Mar. 1996, pp. 301/302 XP000581702.

ART-UNIT: 211

PRIMARY-EXAMINER: Banankhah; Majid

ABSTRACT:

A memory monitor automatically restores the state of physical memory allocation of application programs when they lose and then regain the focus in a multitasking computing environment. The memory monitor monitors the focus of the operating system for changes, such as when the user switches from one application to another. When an application loses the focus, the memory monitor determines and stores the state of physical memory allocation. When the memory monitor detects that the application has re-gained the focus, it re-loads all of the code or data that was in physical memory when the application lost the focus, but had been swapped to secondary storage.

19 Claims, 5 Drawing figures